

**ARQUIVO, REGISTRO E CHAVE** - Um arquivo é uma coleção lógica de registros sequenciais. Eles são mapeados em blocos de disco que possuem um tamanho fixo determinado pelas propriedades físicas do disco e pelo sistema operacional. Os registros podem ter tamanhos fixos ou variáveis e são compostos por campos que representam atributos da entidade representada. Uma chave é uma sequência de um ou mais campos. A chave primária é uma chave que apresenta valores distintos para cada registro do arquivo. A chave secundária difere da primária por permitir duplicações e, desta forma, normalmente identifica um conjunto de registros.

**REPRESENTAÇÃO DE REGISTROS** - Para registros de tamanho fixo, colocamos um atrás do outro. Na exclusão, usa-se a ideia de lista livre, com o endereço inicial no cabeçalho, p/ preencher os espaços excluídos c/ novos registros. Para registros de tamanho variável, podemos colocar um caracter de fim de registro, porém, por problemas de reutilizar um espaço ocupado por um registro apagado, utiliza-se a estrutura slotted-page (FIGURA 1), onde os registros são organizados dentro de um bloco, de forma contínua, a partir do fim do bloco. Outra maneira é preencher com espaços em branco os registros p/ que se tornem fixos. Isso é útil quando a maioria dos registros tem um tamanho próximo do tamanho máximo. Por fim, outra maneira é criar um campo de endereço em cada registro que aponte p/ uma área de overflow quando necessário.

**ORGANIZAÇÃO DE ARQUIVOS** - Dado um conjunto de registros, temos 4 principais maneiras de organizá-los dentro de um arquivo: sequencial, sequencial-indexado, indexado e acesso direto.

**ORGANIZAÇÃO SEQUENCIAL** - A ordem lógica e física dos registros é a mesma e eles estão ordenados por uma chave primária, única, e crescente. São úteis p/ classificação e acesso de um grande número de dados e processamento serial. Consulta-se por pesquisa exhaustiva ou pesquisa binária, esta quando a chave de acesso = chave de ordenação. A inserção pode ser feita na posição adequada movendo um grande número de registros ou em arquivo auxiliar. A exclusão pode ser física ou lógica. A reorganização elimina registros excluídos logicamente e intercala os arquivos auxiliares.

**ORGANIZAÇÃO SEQUENCIAL-INDEXADA** - Quando o volume de acesso torna-se muito grande, temos associado ao arquivo sequencial de dados um arquivo de índice, que é formado por duplas: chave de acesso e endereço no arquivo de dados. O arquivo de índice deve estar fisicamente ordenado. A inclusão de registros deve ser feita na área de extensão, que pode ser implementada com um campo de elo em cada registro ou um campo de elo a cada bloco (alternativa inviável se existirem índices secundários). A exclusão é realizada logicamente. A reorganização do arquivo é necessária quando a área de extensão ficar muito grande, além de reestruturar o arquivo de índices e eliminar registros excluídos logicamente. Esta organização está representada na FIGURA 2.

**ORGANIZAÇÃO INDEXADO** - Quando a necessidade de acessos seriais é reduzida, podemos acessar o arquivo apenas por índices, não sendo necessário manter a sequencialidade física dos registros no arquivo de dados. Dessa forma, a área de extensão é dispensada, porém, os índices devem estar atualizados sempre. Podem existir tantos índices quantas forem as chaves de acesso do arquivo de dados. O índice primário de um arquivo indexado é chamado de índice exhaustivo, pois possui uma entrada p/ cada registro. Os índices secundários são chamados de índices seletivos, p/ cada entrada se relaciona c/ um subconjunto do arquivo de dados. A exclusão pode ser física ou lógica e a alteração de um registro pode implicar na alteração dos índices. A reorganização só é necessária quando os registros excluídos logicamente não são aproveitados pelo algoritmo de inclusão. Esta organização está representada na FIGURA 3.

**ORGANIZAÇÃO DE ACESSO DIRETO** - Há uma relação definida entre o valor da chave primária de um registro e seu endereço. Há dois métodos: no endereçamento direto, por exemplo se o valor da chave varia de 0 a 4000, deve-se gravar 4000 registros, mesmo que alguns em branco, já que cada registro possui um endereço físico pre-definido. É vantajoso quando o número de valores de chave possível é próximo do número real de valores da chave; no endereçamento indireto, todo o registro lógico é mapeado p/ uma área de armazenamento através de uma função hash. Essas áreas podem conter um ou mais registros. O valor gerado pela função hash é usado em um acesso por endereçamento direto. Os seguintes fatores devem ser considerados na definição de uma função hash: tamanho da área (área pequena gera muitas colisões e grande desperdiza espaço), fator de carga (proporção entre o n.º de registros existentes no arquivo e o n.º de registros disponíveis no arquivo) e o tipo de transformação. A consulta de registros é realizada pela geração do endereço pela função hash e pelo método de tratamento de colisões. A exclusão lógica é melhor e a reorganização pode ser feita para eliminá-las.

**ORDENAÇÃO DE ARQUIVOS: PARTICIONAMENTO** - Métodos de geração de partições: classificação interna: lê M registros p/ a memória, ordena-os e grava em uma partição; seleção com substituição: lê M registros p/ a memória; seleciona destes o menor e grava na partição; substitui, no array de mem pelo próximo registro de entrada; caso a chave deste último seja menor do que a chave recém-gravada, considerá-lo congelado; enquanto existirem posições não congeladas, repetir, caso contrário, uma nova partição será aberta e o array de memória começará c/ os registros congelados. O tamanho médio das partições é  $2 \cdot M$ ; seleção natural: lê M registros da memória, seleciona o menor p/ a partição, substitui pelo próximo da entrada e caso ele seja menor do que o recém-gravado, colocar no reservatório (de tamanho M e na mem. secundária) e substituir de novo pelo próximo. Quando o reservatório encher, pegar os valores restantes na memória e jogar na partição, colocar o reservatório no array de memória, esvaziá-lo e abrir nova partição.

**ORDENAÇÃO DE ARQUIVOS: INTERCALAÇÃO** - Todas as partições são abertas e o primeiro registro de cada um é carregado na memória, o menor é gravado na saída e substituído pelo seu sucessor. Usa-se Arvore Binária



de vencedores p/ diminuir as operações de busca (FIGURA 4). Quando o número de arg. é maior do que o suportado, mais fases são necessárias e usa-se a intercalação ótima (FIGURA 5). **ÍNDICES: LISTAS** - útil p/ chaves secundárias, já que uma chave pode retornar muitos valores (registros). **ÍNDICES: Mapeamento de Bits** - útil quando a chave secundária possui um conjunto bem definido de valores. **ÍNDICES: ARVORE B** - útil p/ chave primária. O número mínimo de chaves em um nó é a ordem (D), e o máximo é 2D. O número mínimo de filhos é D+1 e o máximo é 2D+1. A raíz é exceção.

Anderson Pastore Rizzo

• FIGURA 5 - inter. ótima c/ 3 arg. de 100 registros cada.

	ENTRADA			SAÍDA
FASE	ARG 1	ARG 2	ARG 3	Nº DE LEITURAS
1	1:100	2:100	6:200	200
2	3:100	4:100	7:200	200
3	5:100	6:200	8:300	300
4	7:200	8:300	9:500	500
			TOTAL →	1200

$$\text{nº de passos} = \frac{1200}{500} = 2,4$$

• FIGURA 3 - org. indexado

nome	end
Ângela	3
Antônio	6
César	4
Gizele	5
Helena	2
Ivan	1

Índice chave sec.

num	end
1050	6
1075	3
1350	4
1900	5
2000	2
2400	1

Índice chave prim.

O tamanho deve ser um núm. primo não próximo de uma potência de 2.  
 $\text{REHASH} = (\text{hash} + 1) \% \text{tamanho}$   
 FUN. HASH ⇒ resto da divisão inteira da chave pelo tamanho do vetor.

num	nome	idade
1 2400	Ivan	28
2 2000	Helena	22
3 1075	Ângela	22
4 1350	César	24
5 1900	Gizele	27
6 1050	Antônio	21

arg. de dados

num	nome	idade
1 1000	Ademar	23
2 1050	Valcir	27
3 1075	Pedro	21
4 1100	Valdir	19
5 1300	Edoardo	33
6 1350	Helena	25
7 1375	Henrique	34

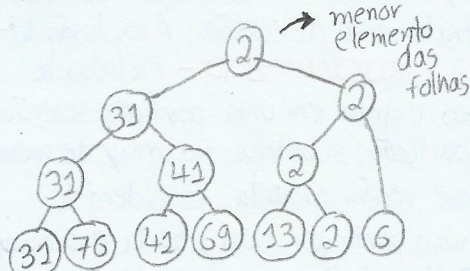
arquivo de dados

: extensão

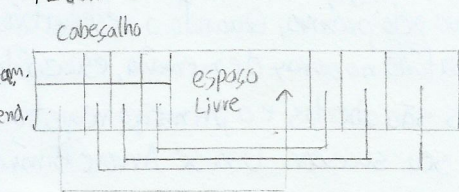
pode conter um campo de exclusão lógica.

pode-se adicionar o campo de elo e exclusão lógica.

• FIGURA 4 - árvore de vencedores



• FIGURA 1 - slotted-page



• FIGURA 2 - org. seq.-indexado

num	end
1100	1
*	3

índice secund.

num	end
1050	1
1100	3
1350	5
*	7

índice primário